SYSTEM AND METHOD FOR SYSTOLIC ARRAY SORTING
OF INFORMATION SEGMENTS

BACKGROUND OF THE INVENTION

5   **1.   Field of the Invention**

This invention relates generally to the transfer of

information segments across a broadband switch and, more particularly,

to a system and method for sorting information segments in a earliest

deadline first (EDF) queue based upon information segment field

10   rankings.

2.   Description of the Related Art

As noted in US Patent 5,608,662 (Large et al.), there are

various situations and various types of communication signal paths in

which it would be desirable to enable a selection of data to be processed

15   and to cause other data to be ignored or perhaps processed differently.

One example is a broadcast data network, i.e., a network wherein data are

transmitted generally to a number of subscriber stations. The data are

subdivided into data set groupings, such as a number of consecutive bytes

typically including leading and/or trailing marker or information bytes or

20   the like, which groupings are termed "packets."

A packet is a sequence of bytes delivered by the

communication line that are rendered distinct from other sequences of

bytes, according to a protocol applied when the bytes are encoded and

decoded. Packet techniques are well known and include, for example, the

25   EtherNet protocol (IEEE standard 802.3) and commercial packet protocols

such as the Synchronous Data Link protocol (SDLC) and X.25. An

external circuit monitors the incoming communication line and determines when the packet begins. Bytes of the packet are then processed until packet reception is complete.

5   Commercially available circuits and interfaces are known for performing the tasks of recognizing the beginning of a packet and processing bytes until complete, for example as known from a byte count, end marker or the like. The generic function of receiving packets is thus known. However, once a packet or sequence of bytes is extracted from the communication network, there are a variety of possibilities as to how the

10   encoded data are to be processed.

The quickest method of sorting information segments is to use a "broadcast"-based sorting method. In this method, an insert or an extract operation can be communicated to every register in the queue simultaneously. Information segments can likewise be inserted or

15   extracted, from register to register, in a single operation. However, the broadcast sorting method has difficulties associated with it. A hardwire connection must be made to each stage. This makes for hardware design problems and associated long interconnect delay. Further, it is difficult to expand or scale such a design because global interconnects do not scale

20   well especially as the technology shrinks to smaller feature sizes..

On the other hand, a systolic array relies upon communications between neighboring stage registers to insert or extract information segments. Thus, the design is flexible and can be scaled. However, the more limited stage-to-stage communication requires

25   relaying information segments stage-to-stage. For example, extraction is done in two steps: One step to extract an information segment from a

stage and one step to fill the empty stage from a neighboring stage so that a next extraction can be done.

It would be advantageous if a system existed for more efficiently handling information segments in a systolic array.

5 It would be advantageous if the stages in a systolic array worked as teams to speed the sorting process.

It would be advantageous if a systolic array sorting system could operate at a higher processing rate. Specifically, it would be advantageous if information to be sorted could be inserted at twice the

10 rate at which it is extracted.

SUMMARY OF THE INVENTION

Accordingly, a method is provided for sorting information segments by field rank in a broadband earliest deadline first queue. The

15 method comprises: inserting information segments, having a field with a sequential rank, at a maximum insertion rate into a temporary register; pairing a temporary register with a corresponding permanent register having the same order in sequence of neighboring registers, forming stages in a sequenced order of stages; systolically moving information

20 segments with lower field ranks to registers higher in a sequenced order of registers; systolically moving information segments with higher field ranks to registers lower in a sequenced order of registers; extracting the information segments, by field rank, at a maximum extraction rate where the maximum insertion rate is equal to twice the maximum extraction

25 rate.

Following the insertion of an information segment into a temporary register in a first stage, the method comprises: comparing the field rank of the information segment in the temporary register to the field rank of an information segment in the first stage permanent register;

5   inserting the information segment having the lower rank into the temporary register of a second stage, higher in order than the first stage; and, moving the information segment having the higher rank into the first stage permanent register.

Following the extraction of an information segment from a

10   register in a first stage, the method comprises: forming a cooperative relationship between the first stage and a second stage, higher in sequence then the first stage; comparing the field ranks of information segments in the cooperating first and second stages; and, following the comparison, moving the information segment with the highest ranking to

15   the permanent register in the first stage.

Additional details of the systolic array sorting method, and a earliest deadline first queue sorting system are described below.

BRIEF DESCRIPTION OF THE DRAWING

20   Fig. 1 is a schematic block diagram illustrating a earliest deadline first queue system for sorting information segments by field rank.

Fig. 2 is a diagram illustrating the timing relationship between the insertion and extraction operations.

25   Fig. 3 is a schematic block diagram illustrating the first stage and the second stage of Fig. 1 in greater detail.

Figs. 4 through 26 illustrate examples of the information segment extraction operations mentioned above in the description of Figs. 1 and 3.

Figs. 27 through 29 illustrate the movement of information

5 segments through independent, or non-cooperating stages.

Fig. 30 is a flowchart depicting a method for sorting information segments by field rank in a earliest deadline first queue integrated circuit of a broadcast switch.

Fig. 31 is a flowchart depicting additional details of the

10 method of Fig. 30.

Fig. 32 is a flowchart depicting an alternate method for sorting information segments in a switching buffer circuit.

DETAILED DESCRIPTION OF THE PREFERRED

15 ## EMBODIMENTS

Fig. 1 is a schematic block diagram illustrating a earliest deadline first queue system for sorting information segments by field rank. The system 100 may operate with a scheduler (not shown), co-located on the same IC with the EDF queue, or on a cooperating IC which

20 supplies the information segment deadlines (ranks). The system 100 comprises a sequence of stages for storing information segments having a field of sequential ranks. A sequence of three stages, a first stage 102, a second stage 104, and a third stage 106 are shown, but the invention is not limited to any particular number of stages. The information segment

25 can be a packet, or section of a packet. In other aspects of the invention,

the information segment includes an address or tag that specifies the location of a packet in memory and the packet's deadline.

Each stage has a port for accepting information segment insertions from a lower sequence stage. As shown, first stage 102 has a

5   port on line 108 to accept an information segment insertion from either a lower sequence stage (not shown), or from the input of the system 100 when the first stage 102 is the lowest sequence stage. Line 110 inserts information segments from the first stage 102 to the second stage 104. Line 112 inserts information segments from the second stage 104 to the

10   third stage 106, and line 114 inserts information segments to a higher sequence stage (not shown), if any.

Each stage includes a port for supplying extracted information segments to a lower sequence stage. As shown, first stage 102 has a port on line 116, depicted as two separate lines, for supplying

15   extracted information segments. The second stage 104 has a port on line 118 to supply extracted information segments. The third stage 106 has a port on line 120 to supply extracted information segments.

Each stage includes a port for accepting extracted information segments from a higher sequence stage. As shown, first stage

20   102 has a port connected to line 118, depicted as two separate lines, to accept extracted information segments. The second stage 104 has a port on line 120 to accept extracted information segments. The third stage 106 has a port on line 122 to accept extracted information segments from a higher sequence stage (not shown), if any.

25   Each stage also includes a port for supplying information segment insertions to a higher sequence stage. As shown, the first stage

102 supplies information segments for insertion on line 110. The second stage 104 supplies information segments for insertion on line 112, and the third stage 106 supplies information segments for insertion to a higher sequence stage (not shown) on line 114, if any.

5   Fig. 2 is a diagram illustrating the timing relationship between the insertion and extraction operations. Each stage accepts information segments inserted at a maximum insertion rate, and supplies information segments extracted at a maximum extraction rate, where the maximum insertion rate is equal to twice the maximum extraction rate. 10 As explained in more detail below, the extraction process is responsive to the field rank of the information segments. As shown in Fig. 2, there is no fixed relationship between the insertions and the extractions. The insertion operation can occur at a maximum rate of once per clock cycle, but an insertion need not occur every cycle. Neither is there a fixed 15 timing relationship between insertion operations. Likewise, the maximum extraction rate is once per two clock cycles, but again, an extraction need not occur every two cycles. Two consecutive extraction operations must be at least two clock cycles apart.

Fig. 3 is a schematic block diagram illustrating the first 20 stage 102 and the second stage 104 of Fig. 1 in greater detail. Each stage includes a temporary register having an input and an output, and a permanent register having an input and an output. As shown, the first stage 102 has a permanent register 200 (P Reg) with an input on line 202 and an output on line 116a. A temporary register 204 (T Reg) has an 25 input on line 108 and an output on line 116b. Likewise, the second stage 104 has a permanent register 206 with an input on line 208 and an output

on line 118a. A temporary register 210 has an input on line 110 and an

output on line 118b.

       The insertion process is relatively simple. A lower sequence

stage inserts information segments into the temporary register of a higher

5   sequence stage. For example, either the temporary register 204 or the

permanent register 200 in the first stage 102 can insert information

segments into the temporary register 210 of the second stage 104.

       The extraction process typically entails a permanent register

in a lower sequence stage receiving information segments extracted from

10   a permanent register in a higher sequence stage. For example, the

permanent register 200 in the first stage 102 can extract an information

segment from the permanent register 206 in the second stage 104.

       Each stage further includes a first, second, and third

comparator. For example, in the first stage 102 a first comparator 220

15   has a first input connected to the temporary register output on line 116b,

a second input connected to the output of the permanent register on line

116a, and an output on line 222 to supply the results of a comparison of

information segment field ranks. A second comparator 224 has a first

input connected to the temporary register output on line 116b, a second

20   input connected to the output of a permanent register in a higher

sequence stage on line 118a, and an output on line 226 to supply the

results of a comparison of information segment field ranks. A third

comparator 228 has a first input connected to the temporary register

output on line 116b, a second input connected to the output of a temporary

25   register in a higher sequence stage on line 118b, and an output on line

230 to supply the results of a comparison of information segment field ranks.

Alternately, each stage could be operated with only two comparators and an additional MUX to selectively supply two sets of the comparator inputs. As seen in the examples below, the first and second comparator are never used at the same time. However, the use of a three comparators, and eliminating the additional MUX, speeds the processing time.

Likewise, in the second stage 104 a first comparator 232 has a first input connected to the temporary register output on line 118b, a second input connected to the output of the permanent register on line 118a, and an output on line 234 to supply the results of a comparison of information segment field ranks. A second comparator 236 has a first input connected to the temporary register output on line 118b, a second input connected to the output of a permanent register in a higher sequence stage on line 120a, and an output on line 238 to supply the results of a comparison of information segment field ranks. A third comparator 240 has a first input connected to the temporary register output on line 118b, a second input connected to the output of a temporary register in a higher sequence stage on line 120b, and an output on line 242 to supply the results of a comparison of information segment field ranks.

Each stage further includes a first multiplexor (MUX) and a second MUX. The first stage 102 has a first MUX 250 with a first input connected to the output of the permanent register in the higher sequence stage on line 118a, a second input connected to the temporary register in

the higher sequence stage on line 118b, a third input connected to the output of the temporary register in the same stage on line 116b, a fourth input connected to the output of the permanent register in the same stage on line 116a, an output connected to the input of the permanent register

5   in the same stage on line 202, and a control input on line 252 to accept MUX selection commands.

A second stage second MUX 254 has a first input connected to the output of the permanent register in the same stage on line 116a, a second input connected to the output of the temporary register in the

10   same stage on line 116b, an output connected to the input of the higher sequence stage temporary register input on line 110, and a control input to accept MUX selection commands on line 256.

Likewise, the second stage 104 has a first MUX 260 with a first input connected to the output of the permanent register in the higher

15   sequence stage on line 120a, a second input connected to the temporary register in the higher sequence stage on line 120b, a third input connected to the output of the temporary register in the same stage on line 118b, a fourth input connected to the output of the permanent register in the same stage on line 118a, an output connected to the input of the

20   permanent register in the same stage on line 208, and a control input on line 262 to accept MUX selection commands.

A second MUX 264 has a first input connected to the output of the permanent register in the same stage on line 118a, a second input connected to the output of the temporary register in the same stage on

25   line 118b, an output connected to the input of the higher sequence stage

temporary register input on line 112, and a control input to accept MUX selection commands on line 266.

Each stage also includes a controller. In the first stage 102, a controller 270 has a first input connected to the output of the first comparator on line 222, a second input connected to the output of the second comparator on line 226, a third input connected to the output of the third comparator on line 230, a fourth input to connected to the output of a second comparator in a lower sequence stage on line 274, a fifth input connected to the output of a third comparator in a lower sequence stage on line 272, a sixth input connected to the output of a first comparator in a higher sequence stage on line 234, a seventh input to accept insertion and extraction commands from a controller in a lower sequence stage on line 276. If the first stage 102 is the lowest sequence stage, then the seventh input of the controller 270 receives instructions from an elements external to the system 100. An eighth input connected to the seventh input of a controller in a lower sequence stage on line 277 to further cooperation between the controllers, a first output connected to the control input of the first MUX on line 252, a second output connected to the control input of the second MUX on line 256, and a third output connected to the seventh input of a controller in a higher sequence stage on line 278.

The controller 270 has a fourth output on line 279a connected to the permanent register 200. The controller 270 erases data in permanent register 200 with commands on line 279a. The controller 270 has a fifth output on line 279b connected to the temporary register 204. The controller 270 holds the current information segment residing in temporary register with commands on line 279b

Likewise, the second stage 104 has a controller 280 with a first input connected to the output of the first comparator on line 234, a second input connected to the output of the second comparator on line 238, a third input connected to the output of the third comparator on line 242,

5   a fourth input connected to the output of a second comparator in a lower sequence stage on line 226, a fifth input connected to the output of a third comparator in a lower sequence stage on line 230, a sixth input connected to the output of a first comparator in a higher sequence stage on line 282, a seventh input to accept insertion and extraction commands from a

10   controller in a lower sequence stage on line 278. The controller 280 has an eighth input connected to the seventh input of a controller in a lower sequence stage on line 276 to further cooperation between the controllers, a first output connected to the control input of the first MUX on line 262, a second output connected to the control input of the second MUX on line

15   266, and a third output connected to the seventh input of a controller in a higher sequence stage on line 284. The controller 280 has fourth and fifth outputs on lines 286 and 288, connected to the permanent register 206 and the temporary register 210, respectively. Commands on these lines hold or erase the information segments in the registers as mentioned

20   above in the description of the first stage 102.

The controllers in the sequence of stages, for example controllers 270 and 280, interact to systolically move information segments with lower field ranks from stages lower in a sequence, to stages higher in sequence (insertion). The controllers in the sequence of stages

25   also cooperate as a team to systolically move information segments with

higher field ranks, from higher sequence stages, to lower sequence stages (extraction).

A stage receives an information segment and a corresponding command message from a lower sequence stage. More specifically, the

5   temporary register of a stage receives an information segment and the controller of a stage receives a command message from a lower sequence stage. For example, the temporary register 210 and the controller 280 of stage 104 receive an information segment and a corresponding command message respectively from the stage 102. A controller also receives

10   comparator results from neighboring stages as well as its own stage. Then, the controller sets the MUXs for appropriate action to move data among neighboring registers. The controller then sends a command message to the controller in the next higher sequence stage to accompany a data movement. In an extraction operation, the higher sequence stage

15   needs to know if the lower sequence stage is having an information segment extracted from permanent register of the lower sequence stage. In an extraction operation the two registers cooperate to fill the "hole" in the lower stage permanent register. Note that the flow of command message is always from a lower sequence stage (controller) to a higher

20   sequence stage (controller). When moving an information segment from a lower sequence stage to a neighboring higher sequence stage, an information segment from either the permanent or the temporary register of the lower sequence stage can be moved only to the temporary register of the higher sequence stage. On the other hand, moving an information

25   segment from a higher sequence stage to a neighboring lower sequence stage is limited to moving an information segment from either the

permanent or the temporary register of the higher sequence stage to the permanent register of the lower sequence stage.

The command message from a lower sequence stage to a higher sequence stage can be one of the following five commands: "idle"

5   when no information segment is moved between a lower sequence stage and a higher sequence stage; "insert" when an information segment is moved from the temporary register of a lower sequence stage to the temporary register of the higher sequence stage; "shift" when an information segment is moved from the permanent register of a lower

10   sequence stage to the temporary register of the higher sequence stage; "extract" when an information segment is moved from the permanent register of a higher sequence stage to the permanent register of the lower sequence stage; "insert and extract" when an information segment is moved from the permanent register of the higher sequence stage to the

15   permanent register of the lower sequence stage while an information segment from the temporary register of the lower sequence stage to the temporary register of the higher sequence stage is moved at the same time.

An information segment is said to be "ordered" when it is

20   placed in a permanent register of a stage. Comparisons are done such that information segments with same field ranks are ordered based on their arrival time. A "shift" operation is used to preserve the right ordering of information segments when moving them from lower sequence stages to higher sequence stages. In a "shift" operation, no comparisons

25   are required as opposed to an "insert" operation. A "shift" operation causes information segments in permanent registers to be shifted from

lower sequence registers to higher sequence registers without any comparisons.

When the temporary register of a first stage receives a shift command from a stage lower in the sequence of stages, and an

5   information segment having the same rank as the information segment in the first stage permanent register, the information segment in the first stage permanent register is shifted to a temporary register in a second stage, higher in the sequence of stages than the first stage.

With the elements of the stages depicted, the insertion

10   operation can be described with greater specificity. When an information segment is inserted into the temporary register 204 in the first stage 102, for example, the first comparator 220 compares the ranks of the information segment inserted into the temporary register 204 and the information segment in the permanent register 200. The first comparator

15   220 supplies the results of the comparison to the controller 270 on line 222. Then, the controller 270 directs the first MUX 250 to supply the information segment with the higher rank to the permanent register 200. The controller 270 also directs the second MUX 254 to supply the information segment with the lower rank to the temporary register 210 of

20   the second stage 104, that is higher in sequence than the first stage 102. The first stage prepares the second stage 104 for the insertion with communication from the first stage controller 270 to the second stage controller 280 through line 278. An insertion operation can "ripple" through to a next stage, if the second stage must empty its temporary

25   register to accept the insertion from the first stage.

When an information segment is extracted, from the first stage permanent register 200 for example, the first stage second comparator 224 compares the rank of the information segment in the first stage temporary register 204 to an information segment in the second

5   stage permanent register 206. The first stage third comparator 228 compares the rank of the information segment in the first stage temporary register 204 to the information segment in a second stage temporary register 210. The second stage first comparator 232 compares the rank of the information segment in the second stage temporary register 210 to the

10   information segment in the second stage permanent register 206. The first stage controller 270, connected to receive the results of the three comparison operations, is then able to control the MUXs to supply the information segment with highest rank to the first stage permanent register 200. The output of a comparator is disregarded if that

15   comparator is comparing one or more registers having no information segment.

The first stage controller 270 sends a message to the second stage controller on line 278 to communicate the extraction operation. The first stage controller 270 directs the first stage first MUX 250 to deliver

20   the highest ranking information segment to the first stage permanent register 200. If the first stage temporary register 204 information segment is not the highest ranked, the first stage controller 270 directs the first stage second MUX 254 to insert that information segment into the second stage temporary register 210.

25   If the second stage permanent register 206 information segment is not the highest ranked, the second stage controller 280

maintains that information segment in the second stage permanent register 206. If the second stage permanent register 206 information segment is extracted and moved to the first stage permanent register 200, a new extraction operation is begun with the second stage 104 cooperating

5   with the third stage 106 (see Fig. 1). An extraction operation can "ripple" through the sequence of stages if "holes" are created in the permanent register. The second stage second comparator 236 compares the rank of the information segment in the second stage temporary register 210 to an information segment in a third stage permanent register (not shown),

10 where the third stage 106 is higher in sequence than the second stage 104. The second stage third comparator 240 compares the rank of the information segment in the second stage temporary register 210 to the information segment in a third stage temporary register (not shown). The third stage first comparator (not shown) compares the rank of the

15 information segment in the third stage temporary register (not shown) to the information segment in the third stage permanent register (not shown).

A first/second stage extraction operation can also result in the insertion of information segments from the first stage 102 to the

20 second stage 104, or the insertion of information segments from the second stage 104, to the third stage 106. Further, as described above, if an information segment is extracted from the second stage permanent register 206, an extraction operation is initiated with the second stage 104 and the third stage 106 cooperating. Thus, the results of an extraction

25 can ripple through the series of stages as long as "holes" are created in the permanent registers.

Figs. 4 through 26 illustrate examples of the information segment extraction operations mentioned above in the description of Figs. 1 and 3. In these examples the information segments have a field known as a deadline. The earliest deadline, the smallest number in these examples, is considered to have the highest rank in the field of deadlines. In some aspects of the invention, the numbering system reaches a maximum value and wraps around (rolls over), however, for simplicity wrap around is ignored in these examples. These examples illustrate some of the basic principles mentioned above in the discussion of cooperating stages. The extraction principles can be summarized as follows:

    1.   Following an extraction, the empty first stage permanent register (a hole) must be filled, if there are information segments in any of the other registers in the first and second stages;

    2.   If an information segment in the first stage temporary register has not been moved to fill a "hole" in the first stage permanent register, it must move to a higher stage to make room for insertions from lower sequence stages;

    3.   Information segments in the second stage permanent register can only move to the first stage permanent register;

    4.   Information segments in the second stage temporary register can:

        a.   fill the hole in the first stage permanent register;

        b.   remain in place if there is no insertion from the first stage temporary register;

    c. be inserted into the third stage temporary register if there is insertion from the first stage temporary register; and

    d. remain in place if the information

5 segments in the second stage registers have the same ranks (deadlines).

    In Fig. 4, an information segment with a deadline of 15 has been extracted from the first stage permanent register. The extraction is indicated by the cross-hatched lines. Since 17 is the smallest number

10 (earliest remaining deadline and highest priority), it is moved from the first stage temporary register to the first stage permanent register. The 20 in the second stage permanent register remains stationary. The 18 in the second stage temporary register remains stationary. This is an example of how an information segment extraction "ripple" dies.

15   In Fig. 5, an information segment with a deadline of 15 has been extracted from the first stage permanent register. Since 17 is the smallest remaining deadline, it is moved from the second stage temporary register to the first stage permanent register. The 20 in the second stage permanent register remains stationary. The 18 in the first stage

20 temporary register is inserted into the second stage temporary register.

    In Fig. 6, an information segment with a deadline of 15 has been extracted from the first stage permanent register. Since 17 is the smallest remaining deadline, it is moved from the first stage temporary register to the first stage permanent register. The 20 in the second stage

25 permanent register remains stationary. It is assumed herein that the lower numbered subscripts have priority (earlier arrival) over higher

numbered subscripts. Subscripts shown in the figures signify arrival order and are just for the purpose of illustration. In the algorithm, a "shift" operation is used to preserve the order of information segments having the same deadline, as mentioned above. The 20 in the second

5   stage temporary register either remains stationary or is inserted into the third stage.

In Fig. 7, an information segment with a deadline of 15 has been extracted from the first stage permanent register. Since 17 is the smallest remaining deadline, it is moved from the second stage temporary

10   register to the first stage permanent register. The 20 in the second stage permanent register remains stationary. The 20 in the first stage temporary register is inserted into the second stage temporary register.

In Fig. 8 an information segment with a deadline of 15 has been extracted from the first stage permanent register. The higher

15   priority 20 is moved from the second stage permanent register to the first stage permanent register. The 21 in the first stage temporary register is inserted into the second stage, and the lower priority 20 in the second stage temporary register is inserted into the third stage.

In Fig. 9 an information segment with a deadline of 15 has

20   been extracted from the first stage permanent register. The higher priority 20 is moved from the second stage permanent register to the first stage permanent register. The 21 in the second stage temporary register is inserted into the third stage, and the lower priority 20 in the first stage temporary register is inserted into the second stage.

25   In Fig. 10 an information segment with a deadline of 15 has been extracted from the first stage permanent register. The 20 is moved

from the second stage permanent register to the first stage permanent register. The 21 in the first stage temporary register is inserted into the second stage temporary register, which is empty.

In Fig. 11 an information segment with a deadline of 15 has been extracted from the first stage permanent register. The 18 is moved from the first stage temporary register to the first stage permanent register. The 20 in the second stage permanent register remains stationary.

In Fig. 12 an information segment with a deadline of 15 has been extracted from the first stage permanent register. A 20 is moved from the second stage permanent register to the first stage permanent register. The 23 in the second stage temporary register either remains stationary or is inserted into the third stage.

In Fig. 13 an information segment with a deadline of 15 has been extracted from the first stage permanent register. The 18 is moved from the second stage temporary register to the first stage permanent register. The 20 in the second stage permanent register remains stationary.

In Fig. 14 an information segment with a deadline of 15 has been extracted from the first stage permanent register. The 18 is moved from the first stage temporary register to the first stage permanent register. The 20 in the second stage temporary register remains stationary.

In Fig. 15 an information segment with a deadline of 15 has been extracted from the first stage permanent register. The 18 is moved from the second stage temporary register to the first stage permanent

register. The 20 in the first stage temporary register is inserted into the second stage.

In Fig. 16 an information segment with a deadline of 15 has been extracted from the first stage permanent register. The 20 is moved

5   from the second stage permanent register to the first stage permanent register.

In Fig. 17 an information segment with a deadline of 15 has been extracted from the first stage permanent register. A 20 is moved from the first stage temporary register to the first stage permanent

10   register.

In Fig. 18 an information segment with a deadline of 15 has been extracted from the first stage permanent register. A 20 is moved from the second stage temporary register to the first stage permanent register.

15   In Fig. 19 an information segment with a deadline of 15 has been extracted from the first stage permanent register. There are no information segments in any of the other first and second stage registers to fill the hole in the first stage permanent register.

In Fig. 20 an information segment with a deadline of 15 has

20   been extracted from the first stage permanent register. A 20 with a higher priority than the 20 in the first stage temporary register is moved from the second stage temporary register to the first stage permanent register. The lower priority 20 is inserted into the second stage.

In Fig. 21 an information segment with a deadline of 15 has

25   been extracted from the first stage permanent register. A 17 is moved from the first stage temporary register to the first stage permanent

register. The 20 in the second stage permanent register remains stationary.

In Fig. 22 an information segment with a deadline of 15 has been extracted from the first stage permanent register. A 17 is moved

5 from the first stage temporary register to the first stage permanent register. A 20 in the second stage temporary register remains stationary, as does the 20 in the second stage temporary register to preserve their ordering based on arrival time.

In Fig. 23 an information segment with a deadline of 15 has

10 been extracted from the first stage permanent register. A 17 is moved from the second stage temporary register to the first stage permanent register. The 20 in the second stage permanent register remains stationary. The 18 in the first stage temporary register is inserted into the second stage.

15 In Fig. 24 an information segment with a deadline of 15 has been extracted from the first stage permanent register. A 20 with the highest priority is moved from the second stage temporary register to the first stage permanent register. The 20 with the intermediate priority in the second stage permanent register remains stationary. The 20 with the

20 lowest priority is inserted from the first stage temporary register to the second stage.

In Fig. 25 an information segment with a deadline of 15 has been extracted from the first stage permanent register. A 20 is moved from the second stage temporary register to the first stage permanent

25 register.

In Fig. 26 an information segment with a deadline of 15 has been extracted from the first stage permanent register. A 20 with a higher priority is moved from the second stage temporary register to the first stage permanent register. The 20 with the lower priority in the

5   second stage permanent register remains stationary.

Figs. 27 through 29 illustrate the movement of information segments through independent, or non-cooperating stages. In Fig. 27, an information segment having a deadline of 19 is inserted into the permanent register, which is empty. This is an example of how an

10 information segment insertion "ripple" dies.

In Fig. 28, the information segment in the temporary register, with the larger (lower priority) deadline, is shifted into a higher sequence stage. In Fig. 29, the information segment in the temporary register, with the smaller deadline (19) is moved to the permanent

15 register. The information segment with the larger deadline (20) in the permanent register is shifted into a higher sequence stage. The insertion operation continues, or ripples to the next stage.

Fig. 30 is a flowchart depicting a method for sorting information segments by field rank in a earliest deadline first queue

20 integrated circuit of a broadcast switch. Although the method is depicted as a sequence of numbered steps for clarity, no order should be inferred from the numbering unless explicitly stated. The method begins with Step 3000. Step 3002 extracts information segments, having a field with a sequential rank, by field rank, at a maximum extraction rate. Step

25 3004 inserts the information segments, at a maximum insertion rate equal to twice the maximum extraction rate. In some aspects of the

invention, inserting information segments in Step 3002 includes inserting information segments into a temporary register.

Step 3006 systolically moves information segments with lower field ranks to registers higher in a sequenced order of registers. Step 3008 systolically moves information segments with higher field ranks to registers lower in a sequenced order of registers. Step 3010 pairs temporary registers with corresponding permanent registers having the same order in a sequence of neighboring registers, forming stages in a sequenced order of stages.

Fig. 31 is a flowchart depicting additional details of the method of Fig. 30. The method begins with Step 3100. Step 3102 systolically moves information segments with lower field ranks to temporary registers higher in a sequenced order of temporary registers. Step 3104 systolically moves information segments with higher field ranks to permanent registers lower in a sequenced order of permanent registers. Step 3106 pairs each temporary register with a corresponding permanent register having the same order in a sequence of neighboring registers, forming stages in a sequenced order of stages.

Step 3108 compares the field ranks of information segments in neighboring registers. Systolically moving information segments in Steps 3102 and 3104 includes moving information segments between neighboring registers in response to the field rank comparisons.

Step 3108a inserts an information segment into a temporary register in a first stage. Step 3108b compares the field rank of the information segment in the temporary register to the field rank of an information segment in the first stage permanent register. Step 3108c

inserts the information segment having the lower rank into the temporary register of a second stage, higher in order than the first stage. Moving information segments between neighboring registers in Steps 3102 and 3104 includes moving the information segment having the higher rank

5   into the first stage permanent register.

Step 3108d extracts an information segment with the highest field rank from each stage. In some aspects of the application, comparing the field ranks of information segments in neighboring registers in Step 3108 includes comparing the field ranks of information segments in

10   consecutive stages of the sequenced order of stages.

Step 3108e, following the extraction of an information segment from a register in a first stage, forms a cooperative relationship between the first stage and a second stage, higher in sequence then the first stage. Step 3108f compares the field ranks of information segments

15   in the cooperating first and second stages. Step 3108g moves the information segment with the highest ranking to the permanent register in the first stage following the comparison.

In some aspects of the invention, comparing the field ranks of information segments in the cooperating stages in Step 3108f includes

20   sub-steps. Step 3108f1 compares the field rank of the information segment in the temporary register of the first stage to the information segment in the permanent register of the second stage. Step 3108f2 simultaneously compares the field rank of the information segment in the temporary register of the second stage to the information segment in the

25   permanent register of the second stage. Step 3108f3 simultaneously compares the field rank of the information segment in the temporary

register of the first stage to the information segment in the temporary register of the second stage. Step 3108f4 ignores the result of a simultaneous comparison of information segments in neighboring registers if one (or more) of the registers is empty.

5   Step 3108h moves the information segment to the second temporary register if the information segment in the first temporary register is not moved into the first permanent register. Step 3108i maintains the information in the second permanent register if the information segment in the second permanent register is not moved into

10   the first permanent register. Step 3108j forms a cooperative relationship between the second stage and a third stage, higher in sequence then the second stage if the information in the second permanent register is moved into the first permanent register.

          In some aspects of the invention, comparing the field ranks of

15   information segments in neighboring registers in Step 3108 includes selecting the first inserted information segment if the information segments have the same field rank. Technically, moving an information segment from a lower stage permanent register to a higher stage temporary register does not require a comparison. Such a move is called a

20   "shift" and not an insert. An "insert" requires comparisons to determine which information segment is earlier (has higher rank), whereas a "shift" does not require any comparison. Previously compared and ordered information segments are moved with a shift operation. Thus, if the information segments in the first stage are received with a shift command

25   and the information segments have the same rank, the information segment in the permanent register is shifted to the temporary register of

the second stage, and the information segment in the first stage temporary register is shifted to the first stage permanent register.

In some aspects, inserting an information segment in Step 3108a includes inserting information segments at a maximum insertion rate. In some aspects of the invention, extracting an information segment in Step 3108d includes extracting the information segments at a maximum extraction rate, where the maximum insertion rate is twice the maximum extraction rate.

Fig. 32 is a flowchart depicting an alternate method for sorting information segments in a switching network buffer circuit. The method begins with Step 3200. Step 3202 compares the field rankings of information segments in neighboring registers. Step 3204 in response to comparing, systolically moves information segments with lower field rankings to temporary registers higher in sequence. Step 3206 in response to comparing, systolically moves information segments with higher rankings to permanent registers lower in sequence. Step 3208 extracts information segments at a maximum first rate. Step 3210 inserts information segments at a maximum second rate, twice the first rate.

Step 3212 pairs each temporary register to a permanent register having the same order in a sequence of registers, forming a sequence of stages. Step 3214 compares the field ranks of the information segments in the temporary and permanent registers of the first stage in response to inserting an information segment into a temporary register in a first stage. Step 3216 in response to extracting an information segment from a permanent register in a first stage, compares the field ranks of an information segment in the first stage temporary to information segments

in a second stage, higher in sequence than the first stage. Information
segments having the same field rank are shifted. The shifting operation
is described in detail above.

A system and method have been provided for sorting
5   information segments in a packet/cell output buffer switch. The invention
is immediately applicable to synchronous optical network (SONET)
communications. However, the invention also has use in to any system
that organizes, stores, and retrieves information. Other variations and
embodiments of the invention will occur to those skilled in the art.

10

WE CLAIM: